Appendix A
Interface Management

This appendix contains state machine descriptions that illustrate a number of behaviors that are described in the accompany patent application.

A.1 SECDED Code Error Tables

This section contains the single error correct and double detect error syndromes for the code used to protect the transaction type for the control symbols. The code is defined in Figure 10A. Errors are detected by generating parity over the syndrome and conditioning that result with an AND of the syndrome to detect the all five bits set case, which is a multiple bit error, not a single bit error.

Table A-1. Single Bit Failure Table

Received SECDED Bit					Received Data Bit						Syndrome
0	1	2	3	4	0	1	2	3	4	5	
F											1 0 0 0 0
	F										0 1 0 0 0
		F									0 0 1 0 0
			F								0 0 0 1 0
				F							0 0 0 0 1
					F						0 1 1 1 0
						F					1 1 0 1 0
							F				1 0 1 1 0
								F			1 1 1 0 0
									F		1 1 0 0 1
										F	1 0 1 0 1

Table A-2. Double Bit Failure Table

Received SECDED Bit					Received Data Bit						Syndrome
0	1	2	3	4	0	1	2	3	4	5	
F	F										1 1 0 0 0
F		F									1 0 1 0 0
F			F								1 0 0 1 0
F				F							1 0 0 0 1
F					F						1 1 1 1 0
F						F					0 1 0 1 0
F							F				0 0 1 1 0
F								F			0 1 1 0 0
F									F		0 1 0 0 1
F										F	0 0 1 0 1
	F	F									0 1 1 0 0
	F		F								0 1 0 1 0
	F			F							0 1 0 0 1
	F				F						0 0 1 1 0
	F					F					1 0 0 1 0
	F						F				1 1 1 1 0
	F							F			1 0 1 0 0
	F								F		1 0 0 0 1
	F									F	1 1 1 0 1
		F	F								0 0 1 1 0
		F		F							0 0 1 0 1
		F			F						0 1 0 1 0
		F				F					1 1 1 1 0
		F					F				1 0 0 1 0
		F						F			1 1 0 0 0
		F							F		1 1 1 0 1
		F								F	1 0 0 0 1
			F	F							0 0 0 1 1
			F		F						0 1 1 0 0
			F			F					1 1 0 0 0
			F				F				1 0 1 0 0

Table A-2. Double Bit Failure Table (Continued)

Received SECDED Bit					Received Data Bit						Syndrome
0	1	2	3	4	0	1	2	3	4	5	
			F					F			1 1 1 1 0
			F						F		1 1 0 1 1
			F							F	1 0 1 1 1
				F	F						0 1 1 1 1
				F		F					1 1 0 1 1
				F			F				1 0 1 1 1
				F				F			1 1 1 0 1
				F					F		1 1 0 0 0
				F						F	1 0 1 0 0
					F	F					1 0 1 0 0
					F		F				1 1 0 0 0
					F			F			1 0 0 1 0
					F				F		1 0 1 1 1
					F					F	1 1 0 1 1
						F	F				0 1 1 0 0
						F		F			0 0 1 1 0
						F			F		0 0 0 1 1
						F				F	0 1 1 1 1
							F	F			0 1 0 1 0
							F		F		0 1 1 1 1
							F			F	0 0 0 1 1
								F	F		0 0 1 0 1
								F		F	0 1 0 0 1
									F	F	0 1 1 0 0

A.2 Link Initialization and Maintenance Mechanism

This section contains the link training and initialization state machine pertaining to sampling window alignment. Training takes place in two circumstances; when coming out of reset and after the loss of reliable input port sampling during system operation.

Link initialization and maintenance actually requires two inter-dependant state machines in order to operate, one associated with the input port and the other with the output port. The two state machines work together to complete the link training. The state machines are intended for a device with an 8-bit port or a device with a 16-bit port. The port can only transition from the "Port Unitialized" state to the "Port OK" state when both halves of the state machine are in their OK state.

A.2.1 Input port training state machine

Figure A-1 illustrates the input port training state machine. Error conditions are only detectable while in the "OK" states (OK and OK_maint_trn). The optional OK_maint_trn state, shaded in Figure A-1, is used to adjust the device input port sampling circuitry during system operation.

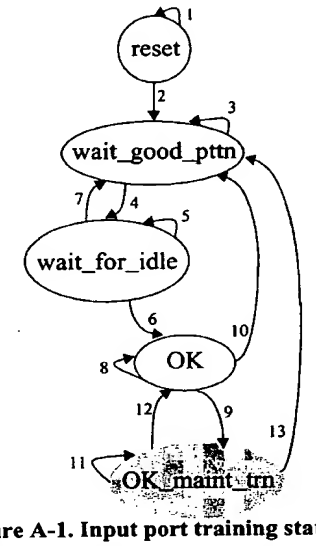


Figure A-1. Input port training state machine

Table A-3 describes the state transition arcs for Figure A-1.

Table A-3. Input port training state machine transition table

Arc	Current State	Next state	cause	Comments
1	reset	reset	Start training condition not met.	Remain in the reset state until the start training condition is met. Typically, this is after reset has been applied to the device and all other necessary initialization activity has completed.
2	reset	wait_good_pttn	Start training condition met.	This state is entered after all initialization activity has completed for the device.
3	wait_good_pttn	wait_good_pttn	The defined training pattern has not been detected yet. Wait for the sampling circuitry to indicate that the defined training pattern has been received and the sampling circuitry is calibrated.	Remain in this state until the defined training pattern is detected.
4	wait_good_pttn	wait_for_idle	Sampling circuitry is calibrated and the defined training pattern has been received.	Upon recognizing the defined training pattern, a 16-bit port can decide whether it's output port needs to be downgraded to drive in 8-bit mode. Request the output port to start sending idle control symbols.
5	wait_for_idle	wait_for_idle	Remain in this state until an exit condition occurs.	In this state, only training patterns and the idle and link-request/send-training control symbols are legal.
6	wait_for_idle	OK	Idle control symbol has been received and the output port is in the "send_idles" state.	This transition indicates that the input port is ready to start receiving packets and other control symbols.
7	wait_for_idle	wait_good_pttn	The input port receives something besides a training pattern, idle, or link-request/send-training control symbol, or the sampling circuitry is no longer calibrated.	Receiving something unexpected or when the sampling circuitry is no longer able to reliably sample the device pins causes both the input port and output port to start restart the training sequence.
8	OK	OK	Sampling circuitry remains calibrated and is not drifting.	This is a functional state in which packets and control symbols can be accepted. Errors are also reported in this state.
9	OK	OK_maint_trn	Sampling circuitry drift.	This transition takes place when the sampling circuitry can still reliably sample the device pins, but adjustment is required to prevent eventual loss of calibration.
10	OK	wait_good_pttn	Sampling circuitry is no longer calibrated.	Both the input port and output port restart the training sequence when the sampling circuitry is no longer able to reliably sample the device pins. This error invokes the error recovery algorithm when the OK state is re-entered to attempt to recover possible lost data.

Table A-3. Input port training state machine transition table

Arc	Current State	Next state	cause	Comments
11	OK_maint_trn	OK_maint_trn	The complete sequence of 256 training patterns has not been received, and the sampling circuitry is still calibrated.	This is a functional state in which packets and control symbols can be accepted. Errors are also reported in this state. In this state, the device adjusts the sampling circuitry when the training patterns are received.
12	OK_maint_trn	OK	The complete sequence of 256 training patterns has been received and the sampling circuitry is still calibrated.	Sampling circuitry has been adjusted.
13	OK_maint_trn	wait_good_pttn	Sampling circuitry is no longer calibrated.	Both the input port and output port restart the training sequence when the sampling circuitry is no longer able to reliably sample the device pins. This error invokes the error recovery algorithm when the OK state is re-entered to attempt to recover possible lost data.

A.2.2 Output port training state machine

Figure A-2 illustrates the output port training state machine. Packets can only be transmitted when both the input port and output port are in their "OK" states (OK and OK_maint_trn for the input port, and OK, OK_send_trn_req and OK_send_trn_pttn for the output port). The 8-bit mode adjustment state for a 16-bit port is heavily shaded in Figure A-2, and is not required for 8-bit ports. The optional OK_maint_trn state, lightly shaded in Figure A-2, is used to adjust the device input port sampling circuitry during system operation, and is associated with the OK_maint_trn state in the input port state machine.

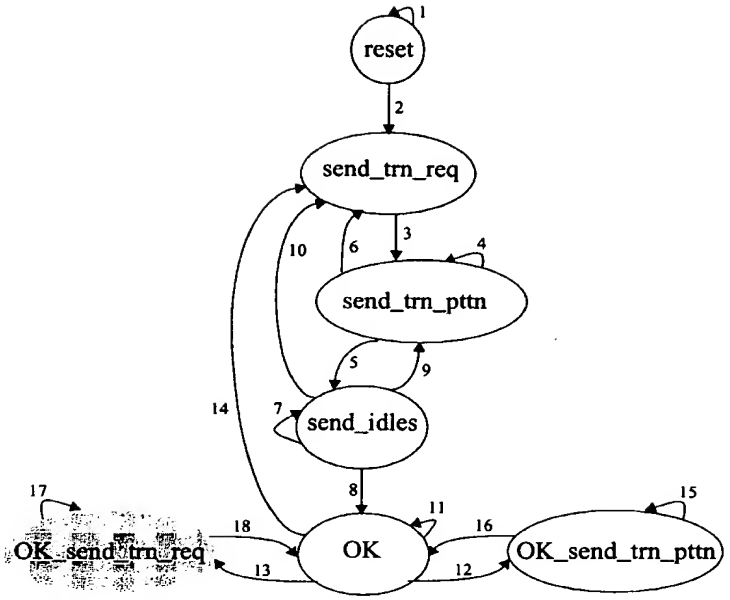


Figure A-2. Output port training state machine

Table A-4 describes the state transition arcs for Figure A-2.

Table A-4. Output port training state machine transition table

Arc	Current State	Next state	cause	Comments
1	reset	reset	Start training condition not met.	Remain in the reset state until the start training condition is met. Typically, this is after reset has been applied to the device and all other necessary initialization activity has completed.
2	reset	send_trn_req	Start training condition met.	This state is entered after all initialization activity has completed for the device. The output port will send a link-request/send-training control symbol
3	send_trn_req	send_trn_pttn	Unconditional transition.	The output port will send 256 iterations of the training pattern
4	send_trn_pttn	send_trn_pttn	The 256 iterations of the training pattern is not completed.	The input port is waiting to calibrate and receive the defined training pattern. The output port is sending training patterns.

Table A-4. Output port training state machine transition table

Arc	Current State	Next state	cause	Comments
5	send_trn_pttn	send_idles	The 256 iterations of the training pattern is completed and the input port has requested to send idle control symbols.	The input port sampling circuitry is calibrated and the input port has received the defined training pattern. In the send_idles state, idle control symbols are sent out on the output port.
6	send_trn_pttn	send_trn_req	The 256 iterations of the training pattern are completed but the input port has not requested to send idle control symbols.	Remain in the send_trn_req - send_trn_pttn loop until the input port sampling circuitry is calibrated and the input port recognizes the defined training pattern and then requests to send idle control symbols. A link-request/send-training control symbol is sent out in state send_trn_req.
7	send_idles	send_idles	Remain in this state until an exit condition occurs.	A reset to the beginning of the training sequence occurs on input port transitions into the wait_good_pttn state. Idle control symbols are sent on the output port in this state.
8	send_idles	OK	The input port is in state wait_for_idle and has received an idle control symbol.	Ready to start sending packets and any control symbol.
9	send_idles	send_trn_pttn	A link-request/send-training is received on the input port.	The output port will send 256 iterations of the of the training pattern as requested.
10	send_idles	send_trn_req	The input port asks for a reset back to the beginning of the training sequence.	Transition to send_trn_req and start over.
11	OK	OK	A link-request/send-training is not received on the input port and the input port does not ask for a reset to the beginning of the training sequence.	This is a functional state in which packets and control symbols are transmitted. Errors are detected and reported in this state.
12	OK	OK_send_trn_pttn	link-request/send-training is received on the input port.	This transition occurs when in the OK state and a training request is received from the attached device.
13	OK	OK_send_trn_req	The input port wants the attached device to send 256 iterations of the training pattern.	This transition occurs when in the OK state and input port sampling circuitry needs to be adjusted, and is associated with the optional input port OK_maint_trn state.
14	OK	send_trn_req	The input port asks for a reset to the beginning of the training sequence.	Transition to send_trn_req and start over. This occurs when the sampling circuitry is no longer able to reliably sample the device pins.

Table A-4. Output port training state machine transition table

Arc	Current State	Next state	cause	Comments
15	OK_send_trn_pttn	OK_send_trn_pttn	The 256 iterations of the training pattern is not completed.	The output port is sending training patterns. Errors are detected and reported in this state. Must send at least one idle control symbol after the 256 iterations.
16	OK_send_trn_pttn	OK	The 256 iterations of the training pattern are completed and followed by at least one idle control symbol.	This is a normal operating case where the attached device requested that we send training patterns.
17	OK_send_trn_req	OK_send_trn_req	Waiting to send the link-request/send-training	Might have to wait for the end of the current packet because link-request control symbols can not be embedded. Errors are detected and reported in this state.
18	OK_send_trn_req	OK	link-request/send-training sent out on the output port as requested by the input port.	Input port is requesting training patterns from the other end to adjust its sampling circuitry.

A.3 Packet Retry Mechanism

This section contains the example packet retry mechanism state machine pertaining to flow control and transaction delivery ordering.

Packet retry recovery actually requires two inter-dependant state machines in order to operate, one associated with the input port and the other with the output port on the two connected devices. The two state machines work together to attempt recovery from a retry condition.

A.3.1 Input port retry recovery state machine

If a packet cannot be accepted by a receiver for reasons other than error conditions, such as a full input buffer, the receiver follows the state sequence shown in Figure A-3.

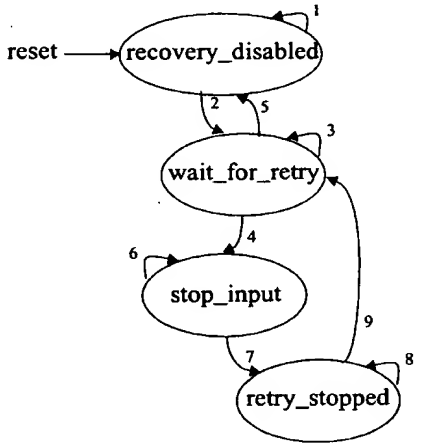


Figure A-3. Input port retry recovery state machine

Table A-5 describes the state transition arcs for Figure A-3. The states referenced in the comments in quotes are the RapidIO 8/16 LP-LVDS defined status states, not states in this state machine.

Table A-5. Input port retry recovery state machine transition table

Arc	Current State	Next state	cause	Comments
1	recovery_disabled	recovery_disabled	Remain in this state until the input port is enabled to receive packets.	This is the initial state after reset. The input port can't be enabled before the training sequence has been completed, and may be controlled through other mechanisms as well, such as a software enable bit.
2	recovery_disabled	wait_for_retry	Input port is enabled.	
3	wait_for_retry	wait_for_retry	Remain in this state until a packet retry situation has been detected.	
4	wait_for_retry	stop_input	A packet retry situation has been detected.	Usually this is due to an internal resource problem such as not having packet buffers available for low priority packets.
5	wait_for_retry	recovery_disabled	Input port is disabled.	
6	stop_input	stop_input	Remain in this state until described input port stop activity is completed.	Send a packet-retry control symbol with the expected ackID, discard the packet, and don't change the expected ackID. This will force the attached device to initiate recovery starting at the expected ackID. Clear the "Port OK" state and set the "Input Retry-stopped" state.
7	stop_input	retry_stopped	Input port stop activity is complete.	
8	retry_stopped	retry_stopped	Remain in this state until a restart-from-retry or restart-from-error control symbol is received or an input port error is encountered.	The "Input Retry-stopped" state causes the input port to silently discard all incoming packets and not change the expected ackID value.
9	retry_stopped	wait_for_retry	Received a restart-from-retry or a restart_from_error control symbol or an input port error is encountered.	The restart_from_error control symbol is a link-request/input-status control symbol. Clear the "Input Retry-stopped" state and set the "Port OK" state. An input port error shall cause a clean transition between the retry recovery state machine and the error recovery state machine.

A.3.2 Output port retry recovery state machine

On receipt of an error-free packet-retry acknowledge control symbol, the attached output port follows the behavior shown in Figure A-4. The states referenced in the comments in

quotes are the RapidIO 8/16 LP-LVDS defined status states, not states in this state machine.

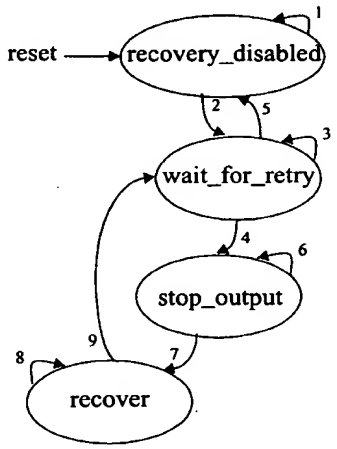


Figure A-4. Output port retry recovery state machine

Table A-6 describes the state transition arcs for Figure A-4.

Table A-6. Output port retry recovery state machine transition table

Arc	Current State	Next state	cause	Comments
1	recovery_disabled	recovery_disabled	Remain in this state until the output port is enabled to receive packets.	This is the initial state after reset. The output port can't be enabled before the training sequence has been completed, and may be controlled through other mechanisms as well, such as a software enable bit.
2	recovery_disabled	wait_for_retry	Output port is enabled.	
3	wait_for_retry	wait_for_retry	Remain in this state until a packet-retry control symbol is received.	The packet-retry control symbol shall be error free.
4	wait_for_retry	stop_output	A packet-retry control symbol has been received.	Start the output port stop procedure.
5	wait_for_retry	recovery_disabled	Output port is disabled.	
6	stop_output	stop_output	Remain in this state until the output port stop procedure is completed.	Clear the "Port OK" state, set the "Output Retry-stopped" state, and stop transmitting new packets.
7	stop_output	recover	Output port stop procedure is complete.	

Table A-6. Output port retry recovery state machine transition table

Arc	Current State	Next state	cause	Comments
8	recover	recover	Remain in this state until the internal recovery procedure is completed.	The packet sent with the ackID value returned in the packet-retry control symbol and all subsequent packets shall be re-transmitted. Output port state machines and the outstanding ackID scoreboard shall be updated with this information, then clear the "Output Retry-stopped" state and set the "Port OK" state to restart the output port. Receipt of a packet-not-accepted control symbol or other output port error during this procedure shall not cause cause a clean transition between the retry recovery state machine and the error recovery state machine.
9	recover	wait_for_retry	Internal recovery procedure is complete.	Re-transmission has started, so return to the wait_for_retry state to wait for the next packet-retry control symbol.

A.4 Error Recovery

This section contains the error recovery state machine link behavior under error.

Error recovery actually requires two inter-dependant state machines in order to operate, one associated with the input port and the other with the output port on the two connected devices. The two state machines work together to attempt recovery.

A.4.1 Input port error recovery state machine

There are a variety of recoverable error types.
The first group of errors are associated with the input port, and
consists mostly of corrupt packet and control symbols. An example of a corrupt packet is a packet with an incorrect CRC. A example of a corrupt control symbol is a control symbol where the second 16 bits are not an inversion of the first 16 bits. The recovery state machine for the input port of a RapidIO link is shown in Figure A-5.

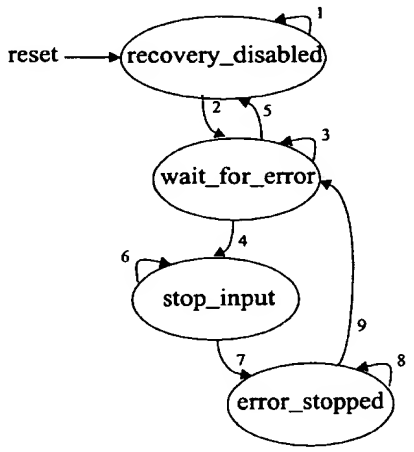


Figure A-5. Input port error recovery state machine

Table A-7 describes the state transition arcs for Figure A-5. The states referenced in the comments in quotes are the RapidIO 8/16 LP-LVDS defined status states, not states in this state machine.

Table A-7. Input port error recovery state machine transition table

Arc	Current State	Next state	cause	Comments
1	recovery_disabled	recovery_disabled	Remain in this state until error recovery is enabled.	This is the initial state after reset. Error recovery can't be enabled before the training sequence has been completed, and may be controlled through other mechanisms as well, such as a software enable bit.
2	recovery_disabled	wait_for_error	Error recovery is enabled.	
3	wait_for_error	wait_for_error	Remain in this state until a recoverable error is detected.	Detected errors and the level of coverage is implementation dependant.
4	wait_for_error	stop_input	A recoverable error has been detected.	An output port associated error will not cause this transition, only an input port associated error.
5	wait_for_error	recovery_disabled	Error recovery is disabled.	
6	stop_input	stop_input	Remain in this state until described input port stop activity is completed.	Send a packet-not-accepted control symbol and, if the error was on a packet, discard the packet and don't change the expected ackID value. This will force the attached device to initiate recovery. Clear the "Port OK" state and set the "Input Error-stopped" state.
7	stop_input	error_stopped	Input port stop activity is complete.	
8	error_stopped	error_stopped	Remain in this state until a restart-from-error control symbol is received.	The "Input Error-stopped" state causes the input port to silently discard all subsequent incoming packets and ignore all subsequent input port errors.
9	error_stopped	wait_for_error	Received a restart_from_error control symbol.	The restart-from-error control symbol is a link-request/input-status control symbol. Clear the "Input Error-stopped" state and set the "Port OK" state, which will put the input port back in normal operation.

A.4.2 Output port error recovery state machine

The second recoverable group of errors described in Section 1.3.5, "Link Behavior Under Error" are associated with the output port, and are control symbols that are error-free and indicate that the attached input port has detected a transmission error or some other unusual situation has occurred. An example of this situation is indicated by the receipt of a packet_not_accepted control symbol. Another example is the receipt of a link-request/ send-training control symbol, which should cause the error recovery procedure to be followed after responding to the request. The state machine for the output port is shown in

Figure A-6.

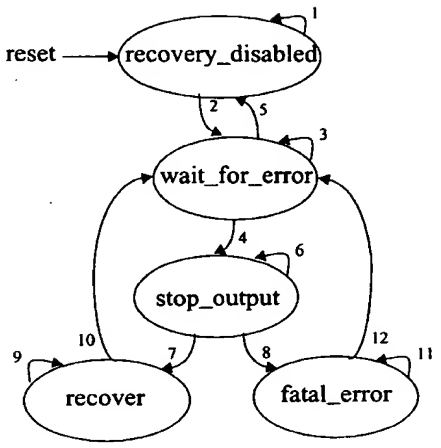


Figure A-6. Output port error recovery state machine

Table A-8 describes the state transition arcs for Figure A-6. The states referenced in the comments in quotes are the RapidIO 8/16 LP-LVDS defined status states, not states in this state machine.

Table A-8. Output port error recovery state machine transition table

Arc	Current State	Next state	cause	Comments
1	recovery_disabled	recovery_disabled	Remain in this state until error recovery is enabled.	This is the initial state after reset. Error recovery can't be enabled before the training sequence has been completed, and may be controlled through other mechanisms as well, such as a software enable bit.
2	recovery_disabled	wait_for_error	Error recovery is enabled.	
3	wait_for_error	wait_for_error	Remain in this state until a recoverable error is detected.	Detected errors and the level of coverage is implementation dependant.
4	wait_for_error	stop_output	A recoverable error has been detected.	An input port associated error will not cause this transition, only an output port associated error.
5	wait_for_error	recovery_disabled	Error recovery is disabled.	

Table A-8. Output port error recovery state machine transition table

Arc	Current State	Next state	cause	Comments
6	stop_output	stop_output	Remain in this state until an exit condition occurs.	Clear the "Port OK" state, set the "Output Error-stopped" state, stop transmitting new packets, and send a link-request/input-status control symbol. Ignore all subsequent output port errors. The input on the attached device is in the "Input Error-stopped" state and is waiting for a link-request/input-status in order to be re-enabled to receive packets. An implementation may wish to time-out several times before regarding a time-out as fatal using a threshold counter or some other mechanism.
7	stop_output	recover	The link-response is received and returned an outstanding ackID value	An outstanding ackID is a value sent out on a packet that has not been acknowledged yet. In the case where no ackIDs are outstanding the returned ackID value shall match the next expected/next assigned ackID value, indicating that the devices are synchronized. Recovery is possible, so follow recovery procedure.
8	stop_output	fatal_error	The link-response is received and returned an ackID value that is not outstanding, or timed out waiting for the link-response.	Recovery is not possible, so start error shutdown procedure.
9	recover	recover	Remain in this state until the internal recovery procedure is completed.	The packet sent with the ackID value returned in the link-response and all subsequent packets shall be re-transmitted. All packets transmitted with ackID values preceding the returned value were received by the attached device, so they are treated as if packet-accepted control symbols have been received for them. Output port state machines and the outstanding ackID scoreboard shall be updated with this information, then clear the "Output Error-stopped" state and set the 'Port OK" state to restart the output port.
10	recover	wait_for_error	The internal recovery procedure is complete.	Re-transmission (if any was necessary) has started, so return to the wait_for_error state to wait for the next error.

Table A-8. Output port error recovery state machine transition table

Arc	Current State	Next state	cause	Comments
11	fatal_error	fatal_error	Remain in this state until error shutdown procedure is completed.	Clear the "Output Error-stopped" state, set the "Port Error" state, and signal a system error.
12	fatal_error	wait_for_error	Error shutdown procedure is complete.	Return to the wait_for_error state even though the output port is shut off.

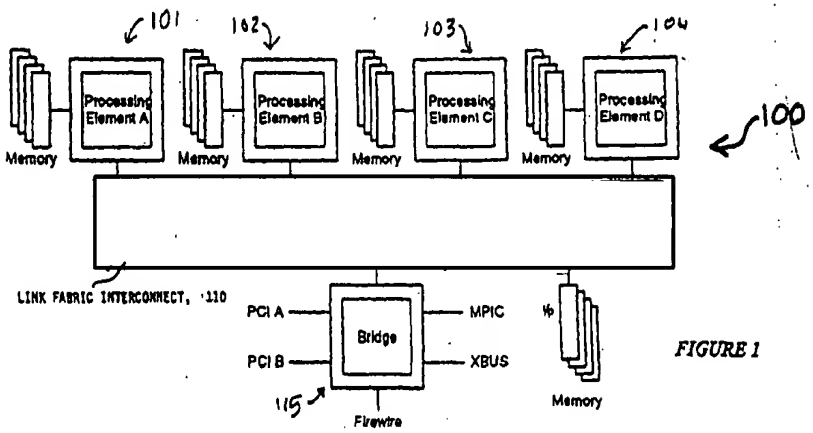


FIGURE 1

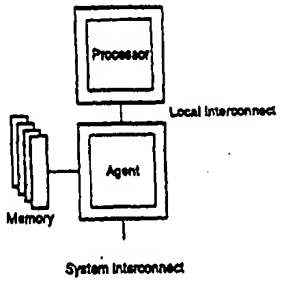


FIGURE 1A

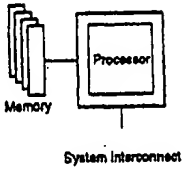


FIGURE 1B

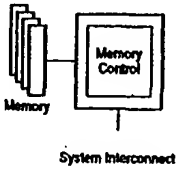


FIGURE 1C

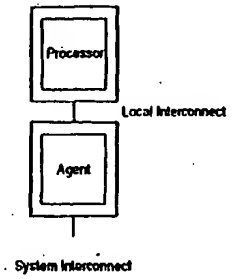


FIGURE 1D

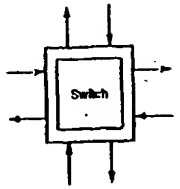


FIGURE 1E

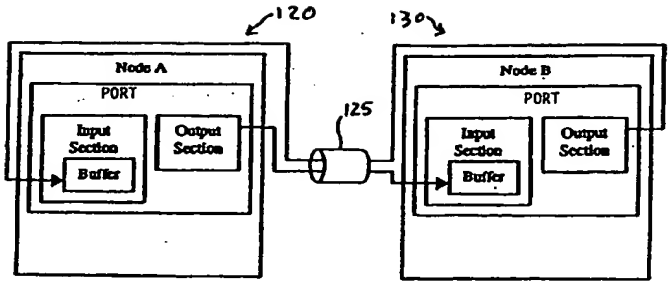


FIGURE 2

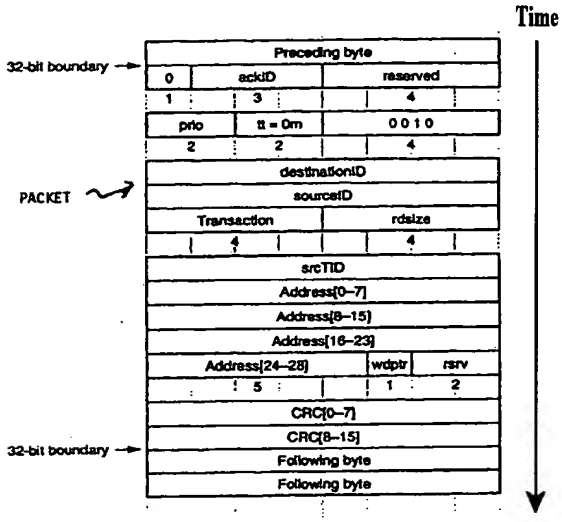


Figure 3

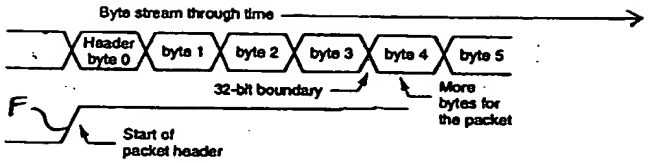


FIGURE 4

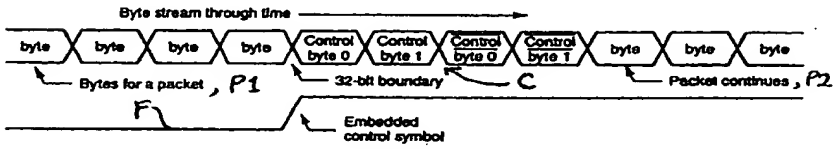


FIGURE 4A

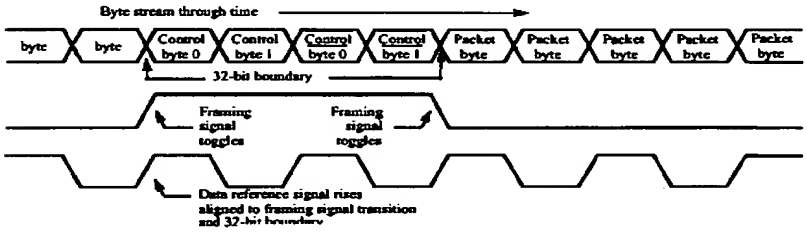


FIGURE 4B

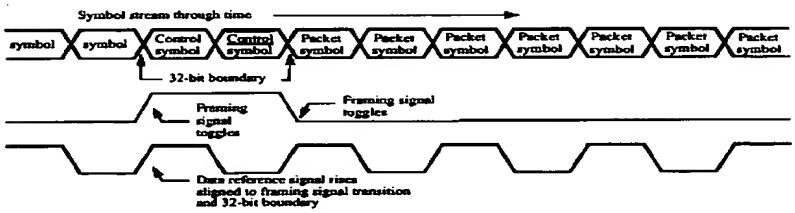


FIGURE 4C

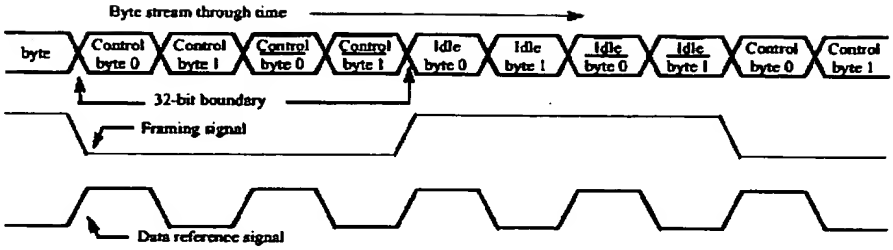


FIGURE 4D

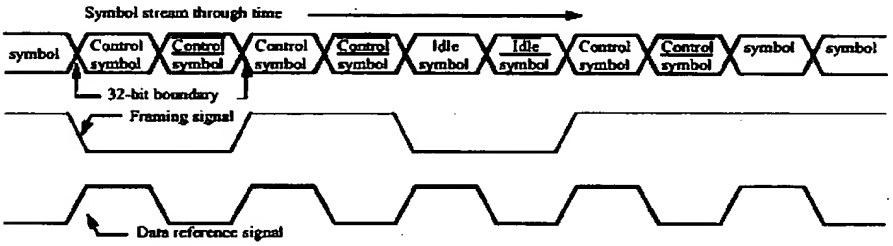


FIGURE 4E

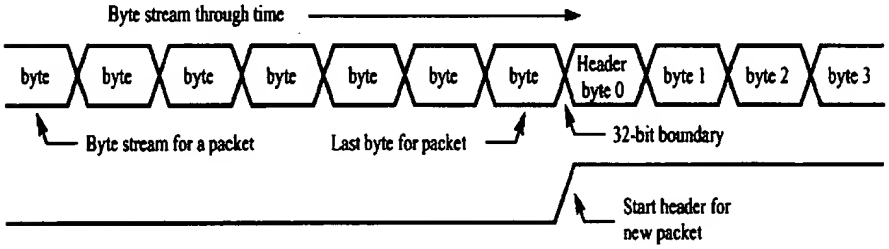


FIGURE 4F

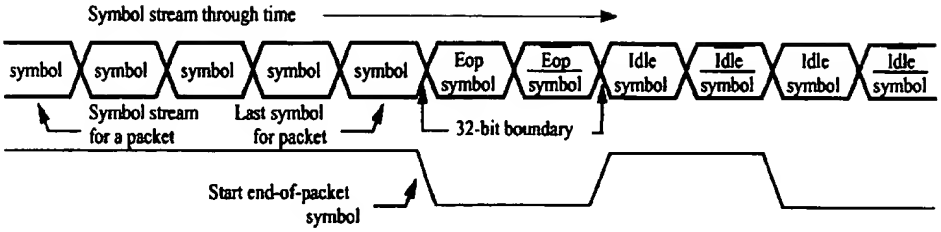
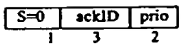
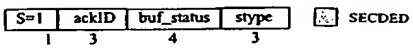

Symbol stream through time
symbol
symbol
symbol
symbol
symbol
Eop symbol
Eop symbol
Idle symbol
Idle symbol
Idle symbol
Idle symbol
Symbol stream for a packet
Last symbol for packet
32-bit boundary
Start end-of-packet symbol

FIGURE 4G



(a)

Packet Physical Layer Fields Format



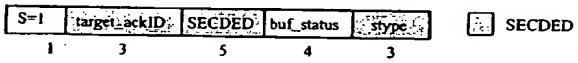
(b)

Control Symbol Physical Layer Fields Format

(c)

Flow Control Fields Bit Stream

FIG. 5



Error Correction Coverage of a Typical Control Symbol

FIG. 6

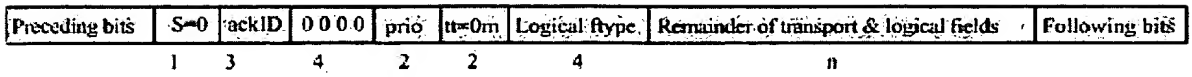


Figure 7. Packet Bit Stream

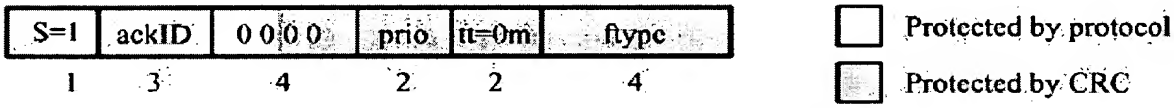


Figure 8. Error Coverage of First Symbol of Packet Header

Control Symbol Single-bit Error Correction

First Symbol Condition	Second symbol condition (re-inverted)	Number of bit mismatches between first and second symbol[1]	Result
No error	No error	0	Continue
Single bit error	No error	1	Correct and continue
No error	Single bit error	1	Continue

[1] This excludes the 4-bit buf_status field since it is not critical to system operation

FIGURE 9A

Table 1-6. Control Symbol Multiple-bit Error Correction

First Symbol Condition	Second symbol condition (re-inverted)	Number of bit mismatches between first and second symbol[1]	Result
Single bit error	Single bit error	0	Correct and continue
Single bit error	Single bit error	2	Correct and continue
No error	Multiple bit error	2	Use first symbol and continue
Multiple bit error	No error	2	Use second symbol and continue
Single bit error	Single bit error	0	Correct and continue
Single bit error	Multiple bit error	3	Correct first symbol and continue
multiple bit error	Single bit error	3	Correct second symbol and continue

[1] This excludes the 4-bit buf_status field because it is not critical to system operation

FIGURE 9B

Table *A* SECDED Encoding

Protection Bit	Data Bit					
	0	1	2	3	4	5
0		x	x	x	x	x
1	x	x		x	x	
2	x		x	x		x
3	x	x	x			
4					x	x

Table *B* Bit Coverage

Data bit	Covered symbol bit
0	ackID[0]
1	ackID[1]
2	ackID[2]
3	stype[0]
4	stype[1]
5	stype[2]

Figure 10

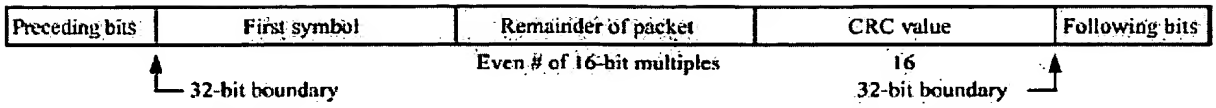


Figure 11A. Naturally Aligned Packet Bit Stream Example 1

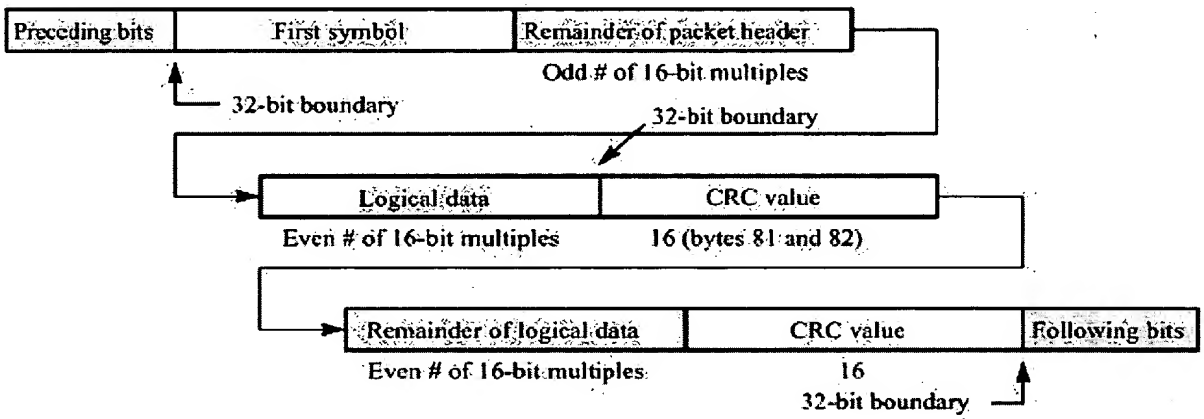


Figure 11B. Naturally Aligned Packet Bit Stream Example 2

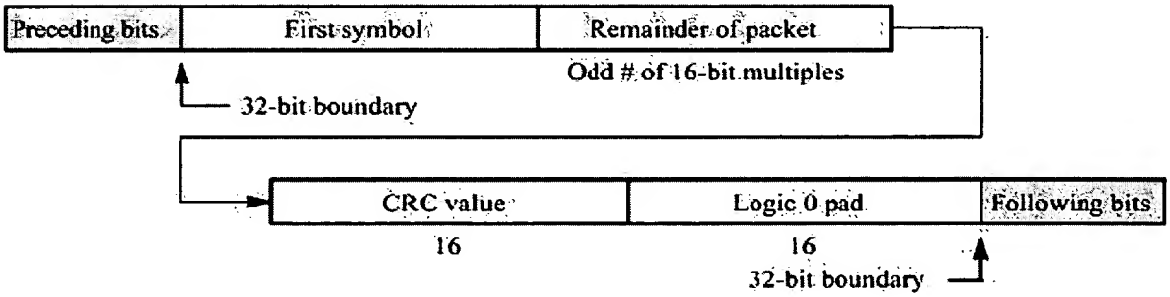


Figure 11C. Padded Aligned Packet Bit Stream Example 1

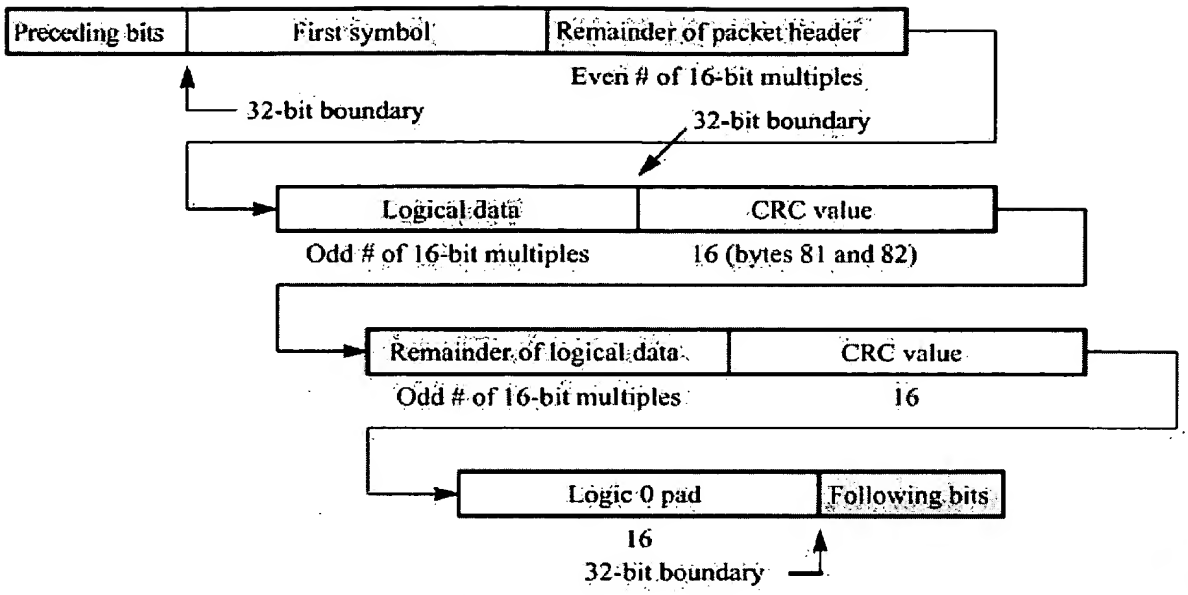


Figure 11D. Padded Aligned Packet Bit Stream Example 2

Parallel CRC Intermediate Value Equations

Check Bit	e00	e01	e02	e03	e04	e05	e06	e07	e08	e09	e10	e11	e12	e13	e14	e15
C00					x	x			x				x			
C01						x	x			x				x		
C02							x	x			x				x	
C03	x							x	x			x				x
C04	x	x			x	x				x						
C05		x	x			x	x				x					
C06	x		x	x			x	x				x				
C07	x	x		x	x			x	x				x			
C08	x	x	x		x	x			x	x				x		
C09		x	x	x		x	x			x	x				x	
C10			x	x	x		x	x			x	x				x
C11	x			x				x				x				
C12	x	x			x				x				x			
C13		x	x			x				x				x		
C14			x	x			x				x				x	
C15				x	x			x				x				x

FIGURE 12

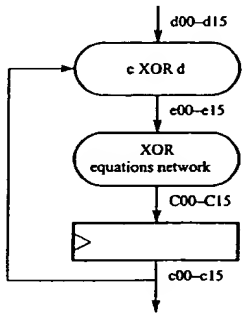


Figure 12 A